

UE 503

L3 MIAGE

TD7 – Couche liaison

A. Belaïd

abelaid@loria.fr

<http://www.loria.fr/~abelaid/>

Année Universitaire 2011/2012





Simulateur CERTA

<http://www.reseaucerta.org/outils/simulateur/>



■ Téléchargement :

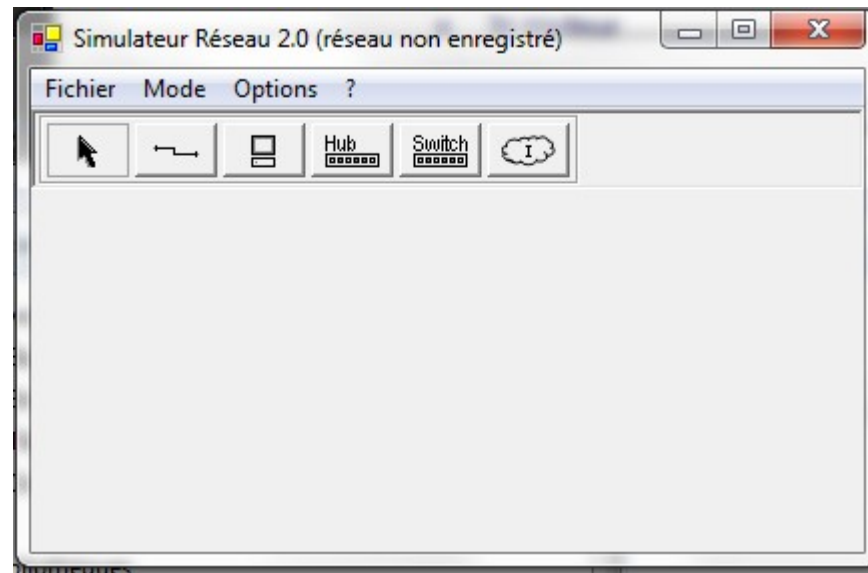
- Nécessité de télécharger une version du framework Microsoft dotNet version 1.1 ou supérieure
 - <http://www.reseaucerta.org/outils/simulateur/>
- et le simulateur :
 - Interface :
 - <http://www.reseaucerta.org/docs/outils/simulateur.zip>
 - Documentation :
 - <http://www.reseaucerta.org/docs/outils/simulateur.pdf>

■ Lancer le simulateur

– Cette interface s'affiche, elle permet de

- Créer un réseau composé de stations de travail, de hubs, de switchs et de câbles
- Simuler le fonctionnement de ce réseau au niveau

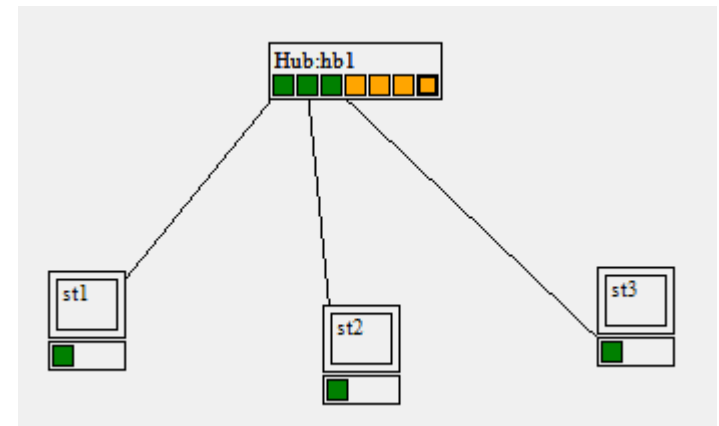
➤ **Ethernet, IP et transport**



■ Mode conception

– Constitution du réseau

- Utiliser les icônes pour créer un réseau
- Tâche 1 : créer un concentrateur
 - Exemples1-cours1\Concentrateur\concentrateur-xml\UnHub.xml
 - Créer 3 stations
 - Créer un hub
 - » les petits carrés sont ceux de la carte réseau
 - Créer des liens en cliquant dans les carrés et joindre les carrés des cartes réseaux
 - » de st1 vers Hub et de Hub vers st2 et st3
 - Passer en Mode : Ethernet pour voir si les connexions sont valides
 - Un point rouge indique que non
 - En survolant les points verts, on lit les adresses provisoires : mac01, mac02 et mac03





Fiche 1



■ L'adressage MAC et la diffusion

– Objectifs

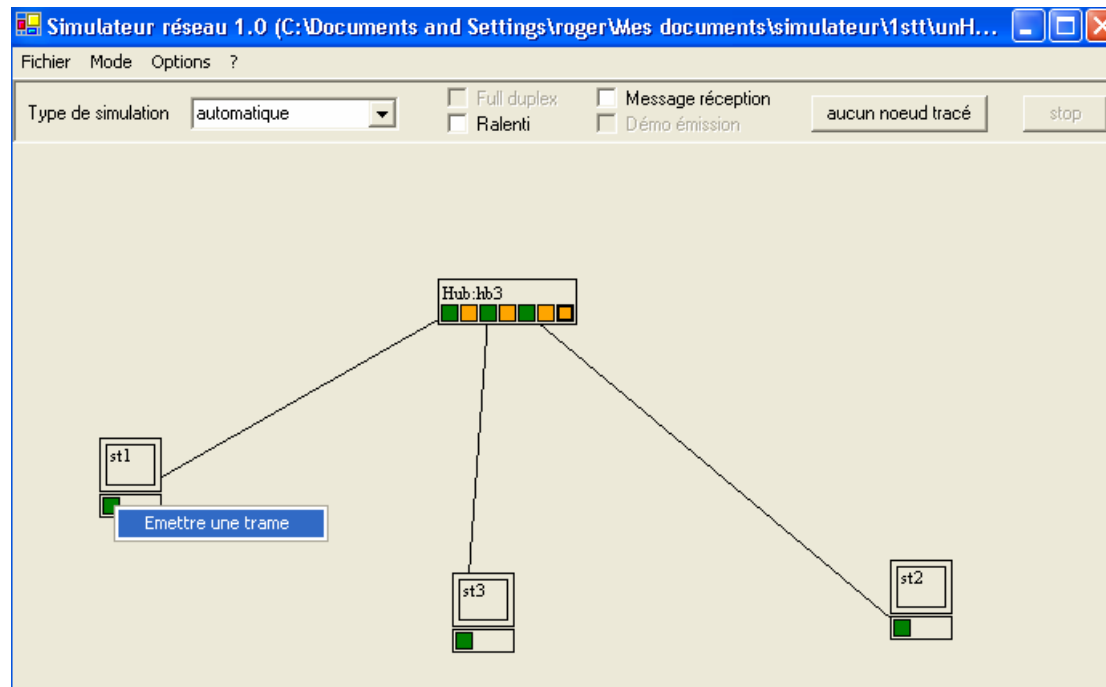
- On s'intéresse dans un premier temps au principe de base de la communication dans un réseau Ethernet « classique » c'est à dire basé sur des concentrateurs (hub)
- On montre comment les cartes réseaux communiquent entre elles, le rôle joué par le concentrateur (la diffusion des trames) et la nécessité des adresses MAC
- On distingue adresse MAC et adresse de diffusion et on distingue aussi diffusion par le concentrateur et trame de diffusion

1. Concentrateur et adresse d'une carte réseau

- *On utilise le fichier unhub.xml., mode automatique avec des trames courtes ou longues selon l'effet qu'on recherche*
- *On va montrer le rôle de chaque élément, la circulation sur le câble et l'envoi à toutes les stations connectées (le concentrateur diffuse)*
- *On va montrer la différence entre la réception de la trame par les cartes et la lecture de la trame (toutes les cartes reçoivent, mais la carte ne lit la trame que si elle reconnaît son adresse ou si il s'agit d'une trame de broadcast)*
- *On peut activer le message de réception ou non pour bien montrer que la carte a lu la trame*
- *Il peut être utile de refaire la démonstration en traçant l'ensemble des postes et le concentrateur pour montrer l'activité de chaque élément*

1. Concentrateur et adresse d'une carte réseau (suite)

- Envoyer une trame unicast à partir de st1 pour st2. *st2 et st3 reçoivent. st2 lit la trame*
 - *Cliquer droit sur st1, sélectionner envoyer une trame*
 - *Sélectionner sur le réseau mac02*
- Envoyer une trame unicast à partir de st2 pour st1. *st1 et st3 reçoivent. st1 lit la trame*
- Envoyer une trame de broadcast à partir de st3. *st1 et st2 reçoivent. st1 et st2 lisent la trame*

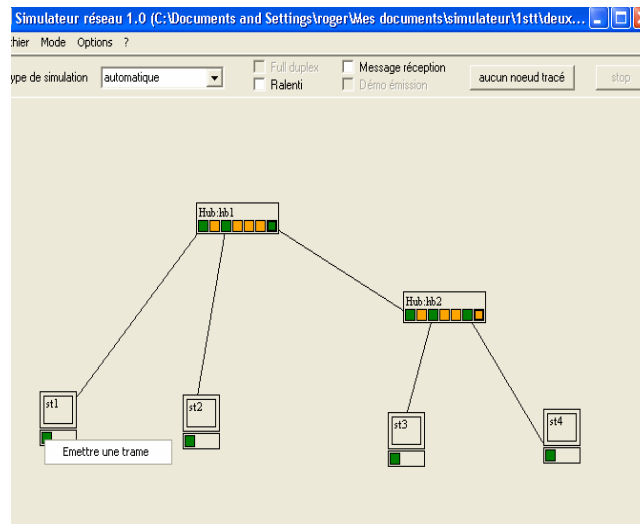


1. Questions-réponses

- Relever les destinataires d'une trame
- Relever les lecteurs d'une trame
- Quelle est la différence de traitement entre une trame unicast et une trame de broadcast ?
- Sur quoi s'appuie une trame pour décider de lire ou de ne pas lire une trame ?

2. Propagation de la diffusion à travers les concentrateurs

- On utilise deuxhub.xml
- Il s'agit de montrer la propagation des messages à travers le concentrateur
- Il faut envoyer d'abord une trame entre deux stations connectées à deux hubs différents et montrer que ce message est diffusé à tous
- Il faut ensuite envoyer une trame entre deux stations connectées à un même hub et montrer que les stations connectées au second concentrateur reçoivent toujours
- Envoyer une trame unicast à partir de st1 pour st3
- Envoyer une trame unicast à partir de st2 pour st1
- Envoyer une trame de broadcast à partir de st4



2. Questions

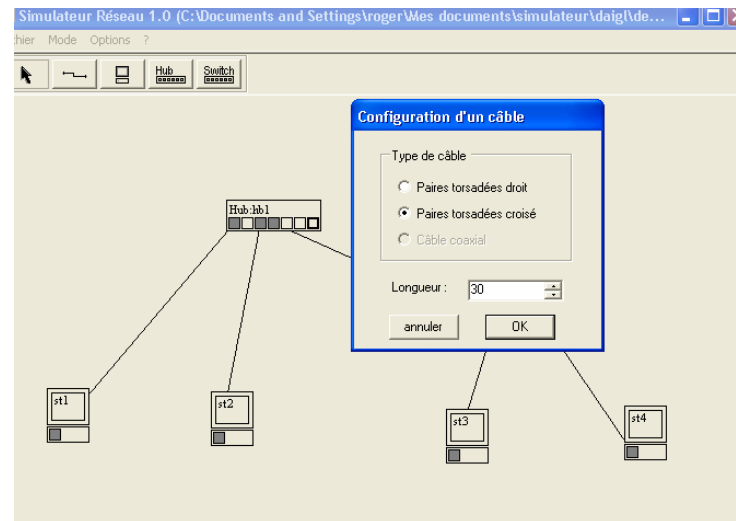
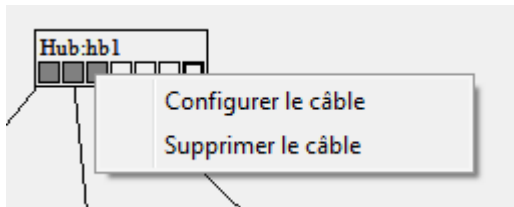
- *Anticiper les résultats et les expliquer*
- *Relever les destinataires d'une trame*
- *Relever les lecteurs d'une trame*
- *Que se passe-t-il si des postes sont éteints ?*
- *Que se passe-t-il si un des concentrateurs est éteint ?*

3. Interconnexion des concentrateurs

- Il est possible de connecter plusieurs hubs entre eux afin de concentrer un plus grand nombre de machines, on parle alors de connexions en cascade (parfois appelé daisy chains en anglais)
- Pour ce faire, il suffit de connecter les hubs à l'aide d'un câble croisé, c'est-à-dire un câble reliant les connecteurs de réception d'une extrémité aux connecteurs de réception de l'autre
- Les concentrateurs sont en général dotés d'un port spécial appelé "**uplink**" permettant d'utiliser un câble droit pour connecter deux hubs entre eux
- Il existe également des hubs capables de croiser ou de décroiser automatiquement leurs ports selon qu'il est relié à un hôte ou à un hub

3. Interconnexion des concentrateurs

- *Faire interconnecter les concentrateurs sans passer par des ports uplink mais en utilisant des câbles croisés (clic droit sur le port)*
 - *un port uplink permet de cascader (empiler) des hubs*
 - *un switch sans uplink est un switch "stand-alone"*





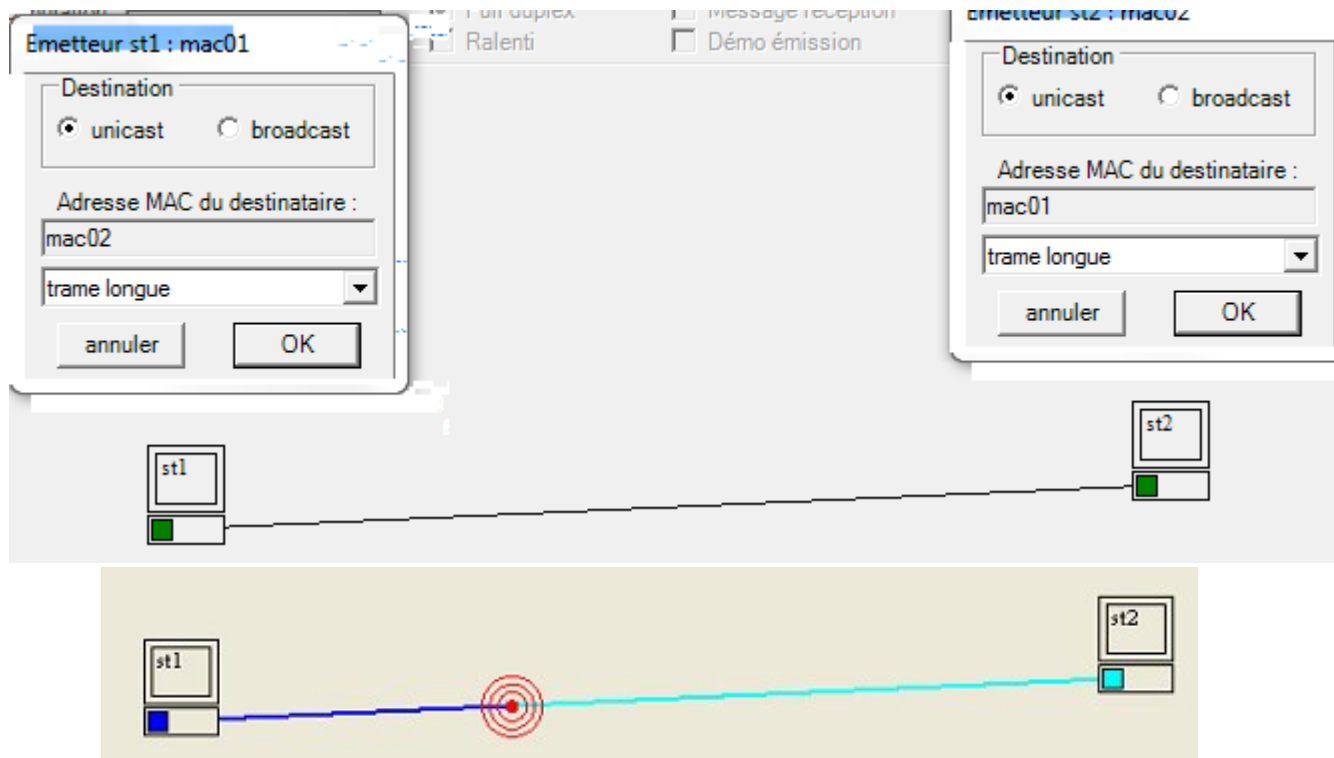
Fiche 2



- La méthode d'accès CSMA/CD
 - On s'intéresse ici au partage du média d'accès
 - Dans Ethernet les stations émettent si le câble est libre
 - Si deux stations émettent au même moment alors que le câble est libre, on parle de collision
 - Le simulateur permet d'illustrer ce phénomène

1. La collision et la méthode d'accès CSMA/CD

- Utiliser le fichier comprendre-reseaux/
sim.xml/2postescablecoaxial
- Utiliser trame réelle
- Envoyer une trame de st1 vers st2 et en même temps une trame de st2 vers st1
- Choisir trame longue pour avoir le temps de faire les deux envois

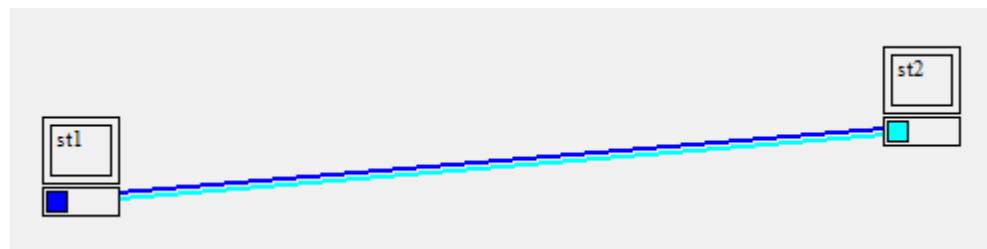
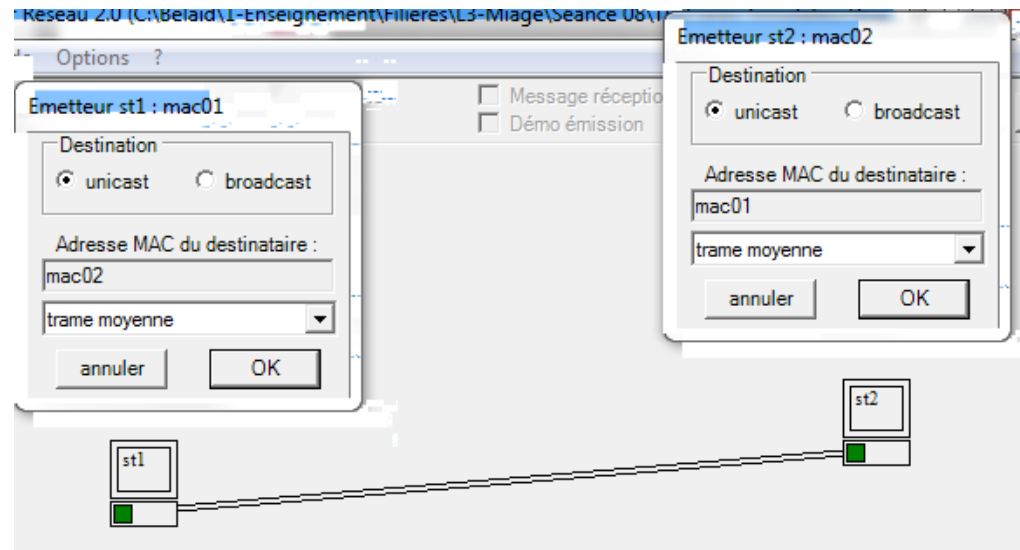


■ Questions

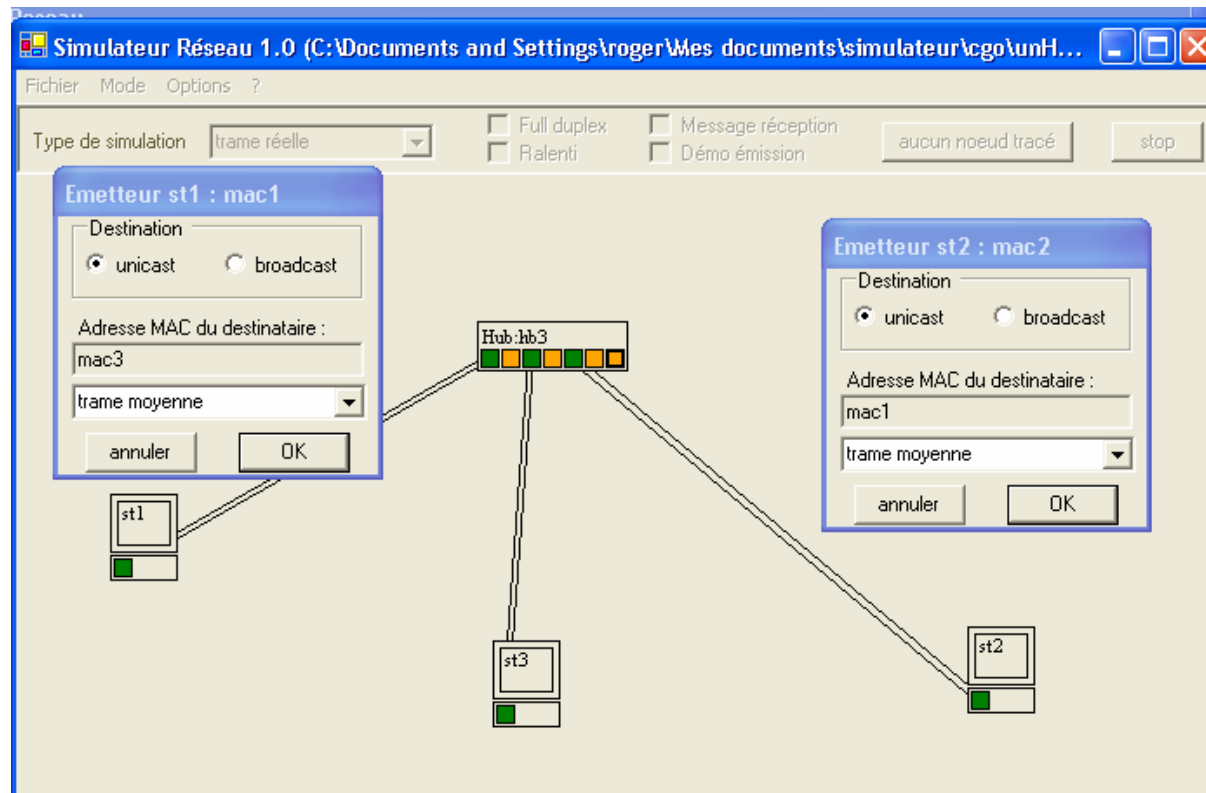
- Que s'est-il passé?
- Les postes s'en sont-ils rendu compte ?
- Qu'ont fait les postes ?

■ Le faire en paire torsadée croisée

- Pour obtenir la paire, configurer le câble en paire torsadée croisée, lancer une trame de chaque poste, alors la paire se double
- On voit qu'il n'y a pas de collision



- Le faire en paire torsadée croisée avec un concentrateur
 - On utilise le fichier unhub.xml
 - On ne configure pas les cables, car déjà fait
 - On montre la collision et la réémission



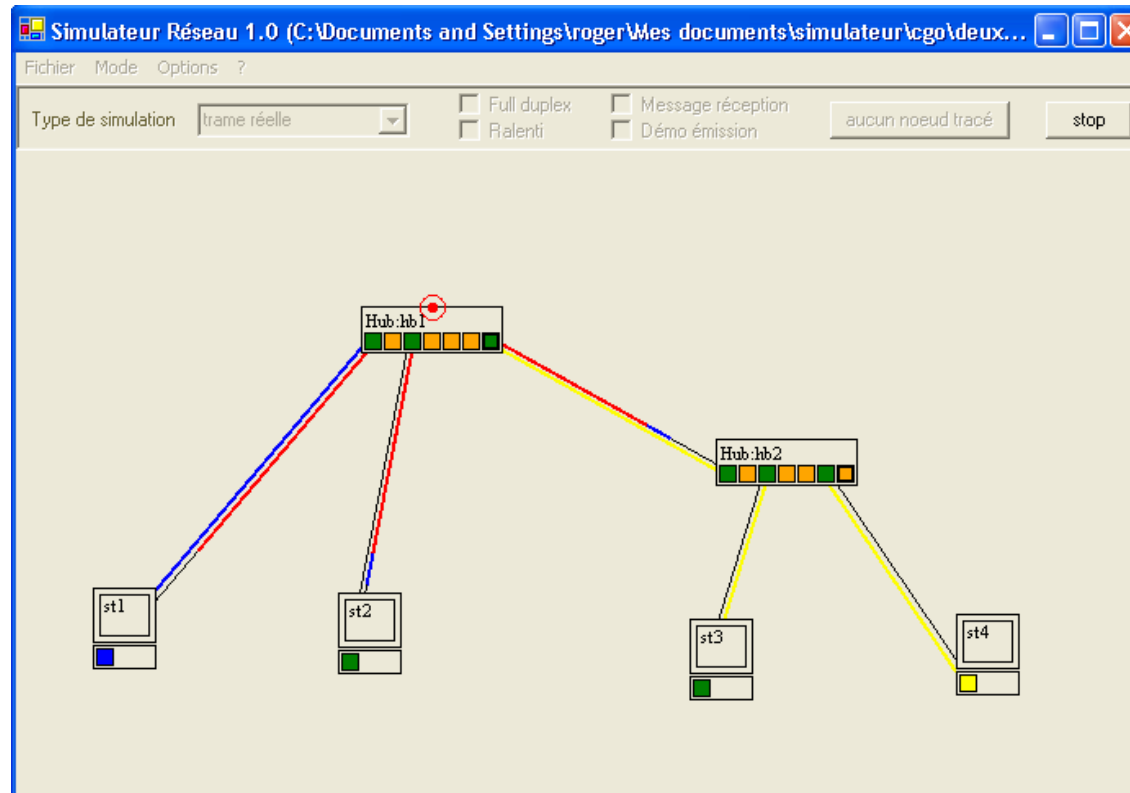
- On alternera les démonstrations « tracées » et « non tracées »
 - Envoyer une trame unicast à partir de st1 pour st3
 - Envoyer en même temps une trame unicast à partir de st2 pour st1

■ Questions

- Où se produit la collision ?
- Quels sont les postes qui détectent la collision ?
- Quels sont les postes qui réémettent une trame ?
- Quand les postes réémettent-ils ?
- Peut-il y avoir de nouveau collision après réémission ?

2. *La propagation de la collision entre les concentrateurs*

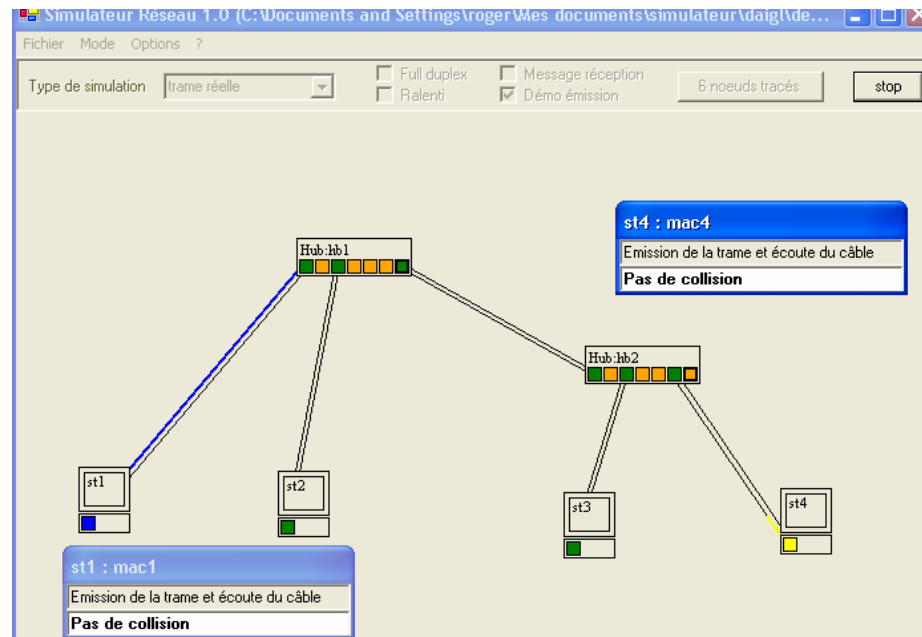
- On utilise le fichier **deuxhub.xml**
- On envoie d'abord une trame entre deux stations connectées à deux concentrateurs différents et on montre que la collision est diffusée à tous
- On envoie ensuite une trame entre deux stations connectées à un même concentrateur et on montre que la collision se propage aussi au second concentrateur
 - Envoyer une trame unicast à partir de st1 pour st3
 - Envoyer en même temps une trame broadcast à partir de st2 pour st1
 - Envoyer une trame de broadcast à partir de st4
 - Envoyer en même temps une trame broadcast à partir de st1



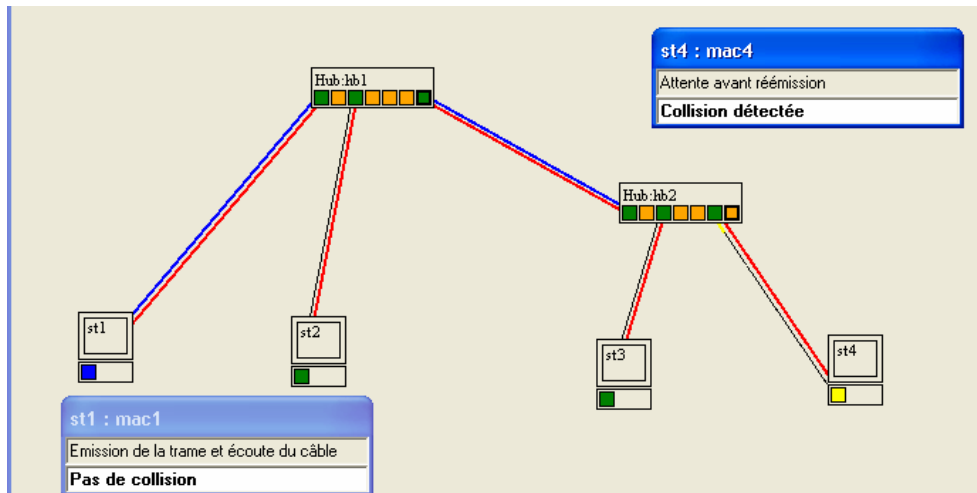
■ Question

- Une trame de broadcast a-t-elle plus de risque de provoquer une collision qu'une trame unicast ?
 - *Non, les deux trames sont propagées de la même façon par le concentrateur*

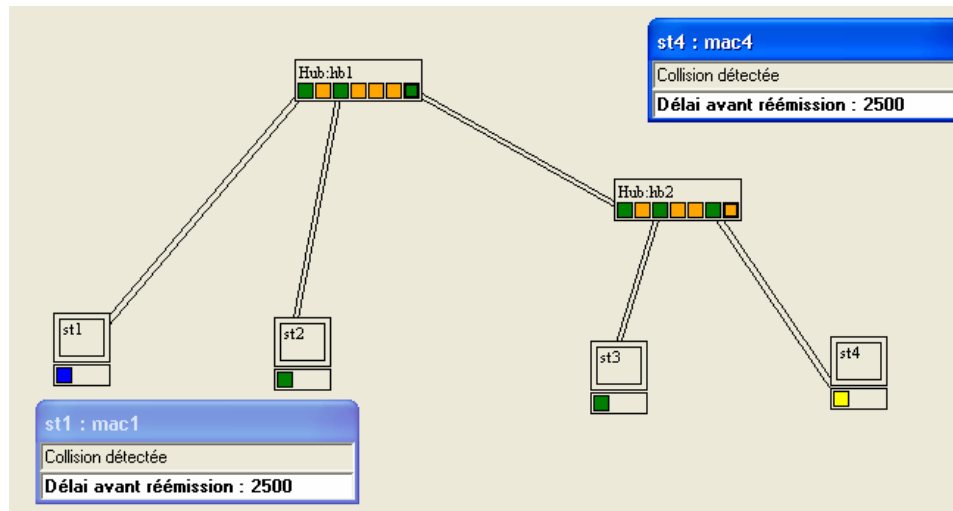
- Pour parfaire la démonstration, on peut utiliser le mode « trame réelle » en démo émission
- Ceci s'obtient en demandant le traçage de différents éléments
- On montre ainsi dynamiquement la gestion des collisions
- St1 et st4 ont envoyé une trame
- Les deux cartes restent à l'écoute sur la paire de réception, pour l'instant elles ne détectent pas de collision



- Une collision s'est produite et se propage par les paires de réception, elle est parvenue à st4 qui l'a détecté mais pas encore à st1.

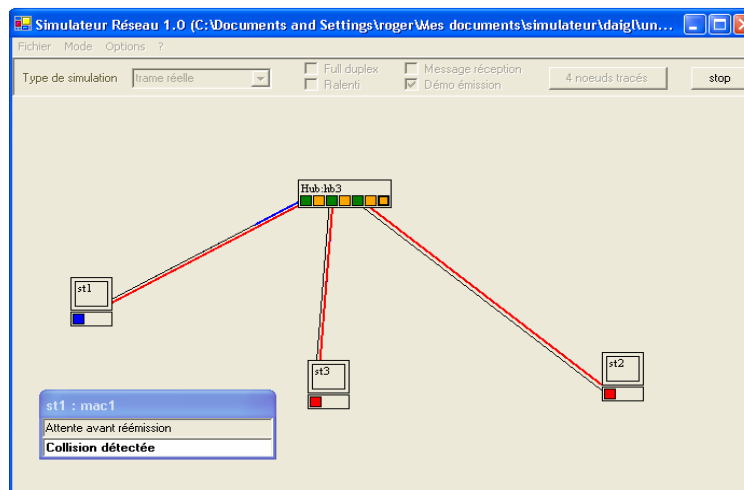


- Les deux cartes ont détecté la collision et ont calculé un délai avant réémission (pas de chance dans cet exemple elles ont calculé le même, il y aura donc une nouvelle collision)



3. Expérimenter le round trip delay

- On utilise le fichier unhub.xml
- On travaille toujours en mode **trame réelle** et démo-émission (des nœuds doivent être tracés)
- On va illustrer le problème du « round trip delay » avec des trames trop courtes dans un premier temps puis avec des câbles trop longs dans un second temps, enfin avec un nombre de répéteurs trop important , et montrer que la collision n'est pas détectée
 - Envoyer une trame unicast à partir de st1 pour st2. *choisir trame moyenne*
 - Envoyer en même temps une trame broadcast à partir de st2 pour st1, choisir trame courte



■ Questions

- Expliquer le comportement de st2
- Refaire cette démonstration avec des trames moyennes mais en affectant 115 comme longueur de câble à st1

4. Construire un réseau composé de cinq concentrateurs

- On peut utiliser cinqhub.xml ou construire son propre réseau
- Montrer les limites de la diffusion (surcharge du réseau) multiplication des collisions, ré-émission incertaine, etc.
- Montrer que lorsque les postes aux extrémités du réseau émettent en même temps (mode trame réelle) la collision n'est pas détectée, indépendamment des longueurs de câbles et de trames, cette fois, c'est le nombre de répéteurs traversés qui pose problème
- D'où la nécessité des commutateurs
- Enfin, montrer qu'entre deux postes avec un câble croisé, les collisions sont impossibles (introduction à l'Ethernet commuté puis au full duplex) : fichier 2postescablecroise.xml et comparer avec le câble coaxial (fichier 2postescablecoaxial.xml)

Fiche 3

■ La commutation

- On veut montrer l'absence de collision et l'absence de diffusion quand on transmet des trames avec des adresses unicast à travers des commutateurs
- Rappel
 - Un commutateur ne diffuse pas les messages à tous les postes
 - Il lit l'adresse du destinataire du message et envoie le message sur le port où est connecté le destinataire

1. La commutation et l'utilisation de l'adresse MAC de l'émetteur

- Utiliser unswitch.xml
- On montre l'apprentissage de la connaissance des adresses MAC des postes connectés
- On montre la différence de traitement entre adressage unicast et adressage multicast
- Enfin on montre l'absence de collision

- Phase d'apprentissage du commutateur (mode automatique)
 - Vider la table des commutateurs
 - Chaque station envoie une fois
 - On envoie une trame de st1 vers st3. On constate que tous les postes reçoivent
 - On envoie une trame de st3 vers st1. Seul st1 reçoit
 - On renvoie une trame de st1 vers st3. Seul st3 reçoit
 - On envoie une trame de st1 vers st2. st2 et st3 reçoivent
 - St2 envoie une trame à st3. st3 reçoit
 - St1 envoie une trame à st2. Seul st2 reçoit. A ce stade le commutateur a associé à chacun de ses ports une adresse de carte réseau
 - On peut continuer à envoyer des trames unicast des différents postes et donc bien insister sur la différence avec un concentrateur
 - St1 envoie une trame de broadcast. Tous les postes reçoivent. Le commutateur diffuse les trames de diffusion et les trames pour lesquelles il n'a pas encore associé de port à l'adresse destinataire de la trame

■ A faire

- Relever les destinataires d'une trame
- Relever les lecteurs d'une trame

■ Questions

- Pourquoi lors du deuxième envoi, seul st1 a été destinataire de la trame ?
- Que faut-il pour que le commutateur sache où se trouvent tous les postes ?
- Le commutateur peut-il ignorer la position d'une station ?
- Que fait un commutateur quand il reçoit une trame dont il ne connaît pas l'adresse du destinataire ?
- Que fait un commutateur quand il reçoit une trame de diffusion ?

- On utilise le mode pas à pas et trace le commutateur
 - *St1 émet une trame vers st3.* On montre la décision prise par le switch

The screenshot shows a network simulator window titled "Simulateur Réseau 1.0". The interface includes a menu bar with "Fichier", "Mode", "Options", and "?". Below the menu bar, there are several controls: a dropdown menu for "Type de simulation" set to "pas à pas", checkboxes for "Full duplex", "Message réception", "Ralentir", and "Démonstration", and buttons for "1 noeud tracé" and "stop".

The main workspace displays a network diagram with three nodes: "st1", "st2", and "Switch:sw1". "st1" is connected to "Switch:sw1" by a blue line, and "st2" is connected to "Switch:sw1" by a black line. A blue box highlights the "sw1" node, which is a switch with five ports. A configuration window for "sw1" is open, showing a table of destinations for port 3.

The configuration window for "sw1" includes a "Sélectionner dest. ds table" field and a table with the following data:

Adresse	Port	TTL
mac1	1	Maximum
mac3	3	Moyen

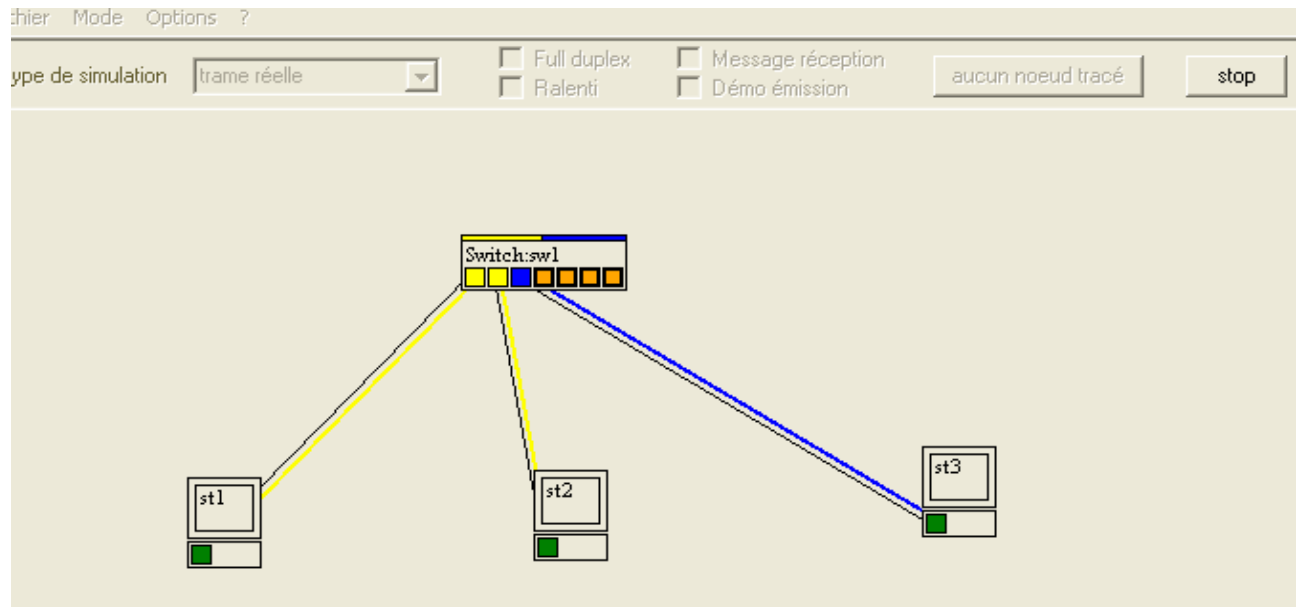
■ Questions

- Comment est construite la table ?
- Qu'est ce que le TTL ?
- A quoi sert le TTL ?
- Que se passe-t-il lorsque le TTL est égal à zéro ?
- Quand le TTL est-il mis à jour ?

■ Absence de collisions

– On utilise le mode trame réelle

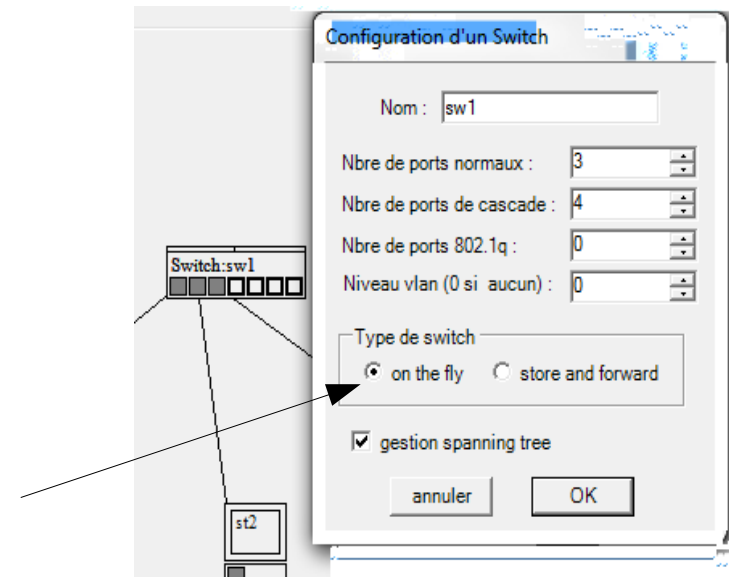
- St1 envoie une trame et st2 envoie une autre trame en même temps (mode manuel), unicast ou broadcast peu importe
- Il n'y a pas de collision
- On voit au-dessus du commutateur les bandes utilisées pour stocker les trames, de manière à éviter leur collision



- Questions

- Pourquoi n'y a-t-il pas de collisions ?

- On configure le commutateur en type « on the fly » et on passe en mode « automatique »
- St1 envoie une trame à st2
 - Il n'y a pas de stockage de la trame, la trame est commutée dès que le commutateur lit l'adresse du destinataire dans la trame
- On configure le commutateur en type « on the fly » et on revient en mode « trame réelle » en traçant tous les nœuds et en cochant «démonstration»
- *St1 envoie une trame à st3. st3 envoie une trame au même moment.* Bien qu'il n'y ait pas de stockage de la trame, il y a écoute du câble avant réémission de la trame



- **Interconnexion de commutateurs (fiche 3)**
 - *On utilise deuxswitvh.xml en mode automatique*
 - *Il s'agit de montrer la gestion de la commutation entre deux commutateurs*
 - *On distinguera encore une fois trame unicast et trame de broadcast.*
 - *On envoie une trame de st1 à st2 connecté au même commutateur. St2 et st3 reçoivent car sw1 a transmis à sw2*
 - *On envoie une trame de st2 à st1 seul st1 reçoit*
 - *On envoie une trame de st1 vers st3. St2 et st3 reçoivent*
 - *On envoie une trame de st2 vers st3. St1 et st2 reçoivent*
 - *On envoie une trame de st3 vers st2. Seul st2 reçoit*
 - *On envoie une trame de st1 vers st3. Seul st3 reçoit. A ce stade les deux commutateurs ont associé à chacun de leur port une adresse de carte réseau*
 - *Envoyer une trame de broadcast à partir de st3*
 - *Envoyer une trame de broadcast à partir de st1*

■ *Questions*

- Pourquoi sw1 transmet-il la trame à sw2 lors du premier envoi ?
- Pourquoi sw1 ne transmet rien à sw2 lors du deuxième envoi ?
- Pourquoi st2 et st3 reçoivent-ils lors du 3eme envoi alors que st2 a déjà été destinataire d'une trame ?

Fiche 4

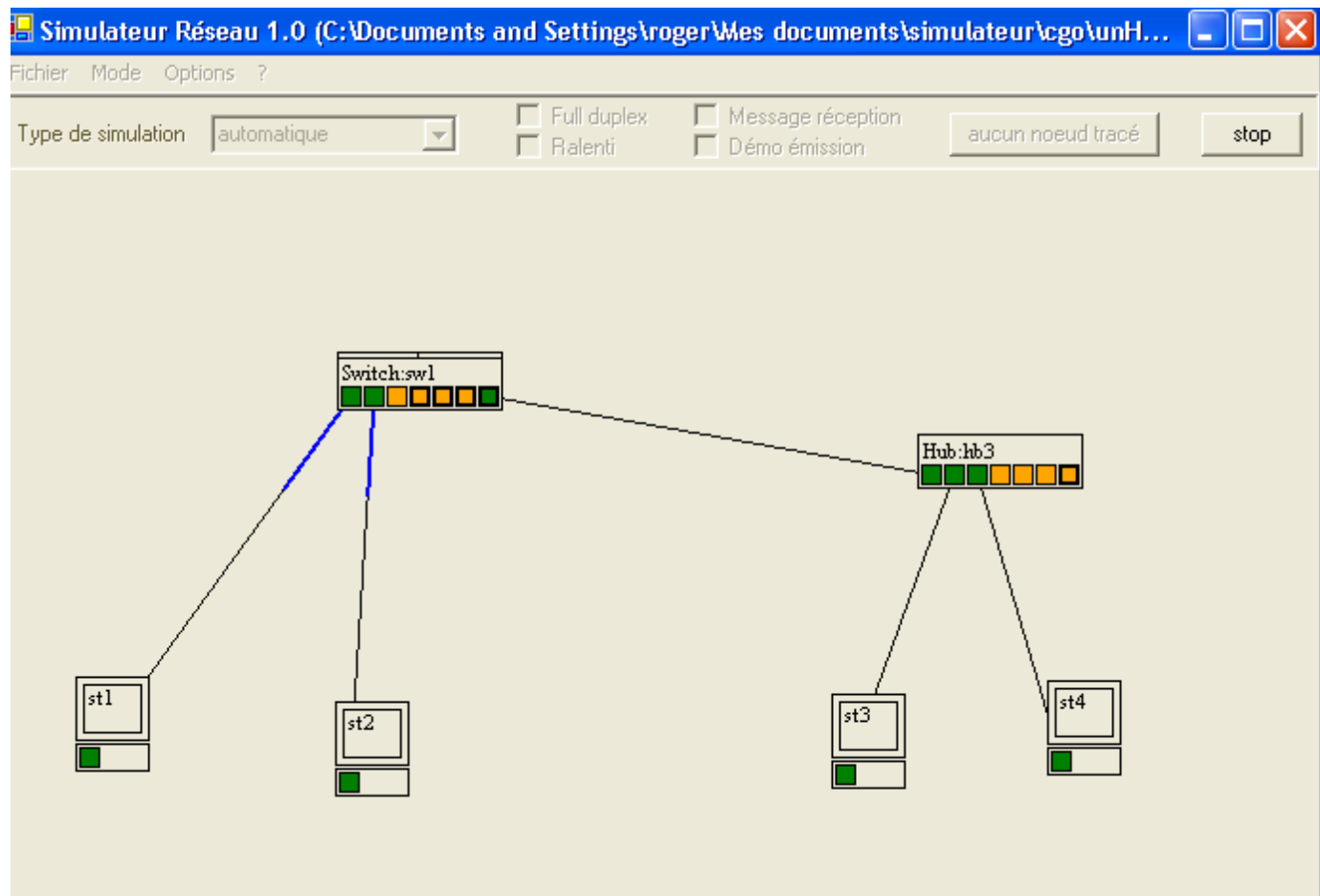
- **Le commutateur comme matériel fédérateur**
 - Il s'agit de montrer la construction d'une architecture complexe en utilisant le commutateur comme matériel fédérateur
 - On veut monter l'interconnexion de commutateurs avec des concentrateurs

■ 1. L'interconnexion des commutateurs et des concentrateurs

- On utilise le fichier unhubunswitch.xml
- On montre la diffusion dans le concentrateur et la commutation dans le commutateur.
- Phase d'apprentissage du commutateur
 - On envoie une trame de st1 à st2 connecté au même commutateur. *St3 et st4 reçoivent car sw1 a transmis à hb3*
 - On envoie une trame de st2 à st1 *seul st1 reçoit.*
 - On envoie une trame de st1 vers st3. *St2 st3 et st4 reçoivent.*
 - On envoie une trame de st2 vers st3. *St3 et st4 reçoivent.*
 - On envoie une trame de st3 vers st2. *St2 et st4 reçoivent.*
 - On envoie une trame de st4 vers st1. *St3 et st1 reçoivent.*
 - On envoie une trame de st4 vers st3. *st3 reçoit. Mais hb3 transmet bien au commutateur. A ce stade le commutateur connaît toutes les adresses.*

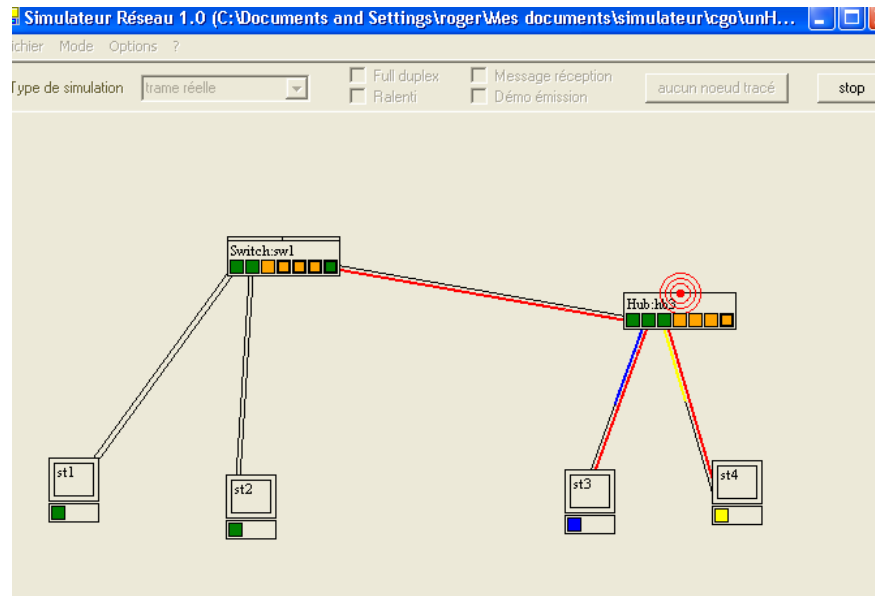
■ 1. L'interconnexion des commutateurs et des concentrateurs

– Schéma



■ 1. L'interconnexion des commutateurs et des concentrateurs

- On utilise le mode trame réelle pour provoquer une collision dans le concentrateur et observer comment réagit le commutateur
 - On envoie une trame de st3 vers st1
 - On envoie en même temps une trame de st4 vers st2. *Il n'y a pas de propagation de la collision par le commutateur (en réalité c'est parce que le commutateur est de type « store and forward » ce qui est généralement le cas).*



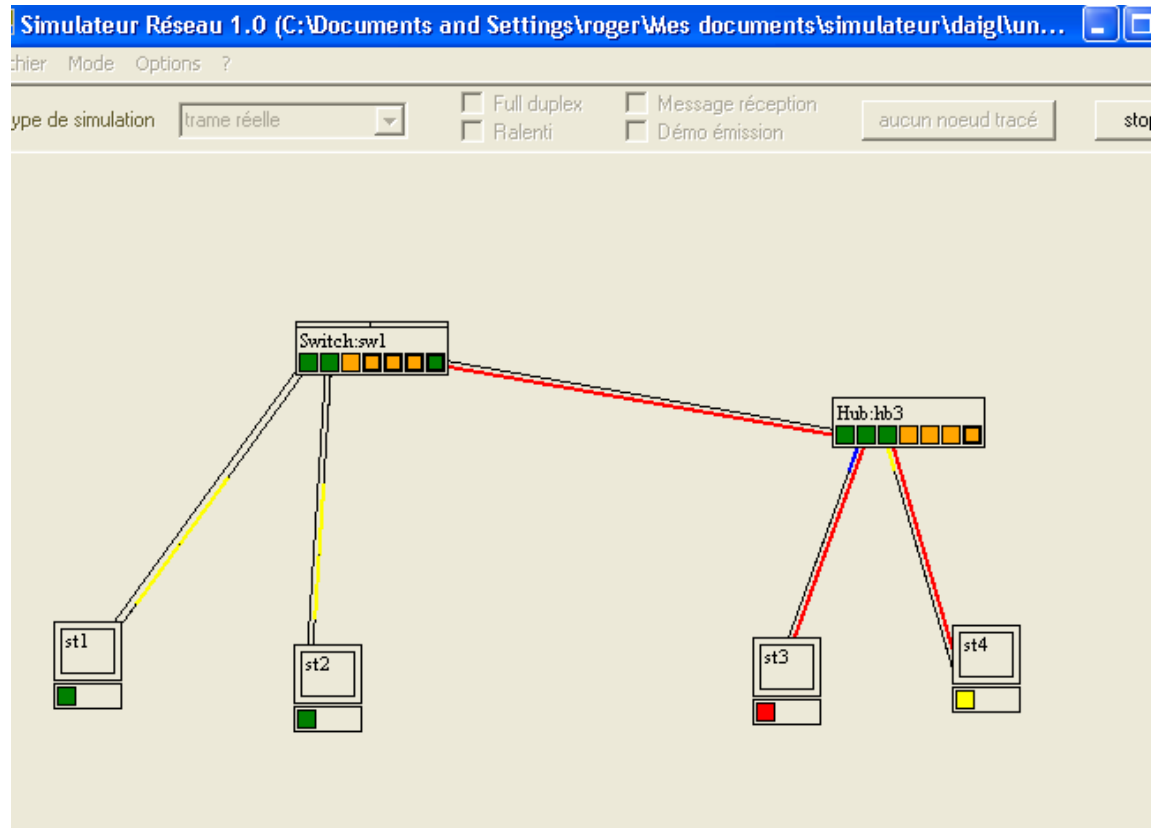
■ 1. L'interconnexion des commutateurs et des concentrateurs

– A faire

- Relever les destinataires de la trame
- Relever les lecteurs de la trame
- Anticiper ou expliquer les résultats. *A ce stade, vous devez pouvoir prévoir le comportement des deux matériels*

- 1. L'interconnexion des commutateurs et des concentrateurs
 - On configure le commutateur en type « on the fly »
 - On reste en mode trame réelle pour provoquer une collision dans le concentrateur et observer comment réagit le commutateur
 - On envoie une trame de st3 vers st1
 - On envoie en même temps une trame de st4 vers st2
 - *Il n'y a pas de propagation de la collision par le commutateur par contre on peut constater que la trame a commencé à être retransmise alors qu'elle a provoqué une collision*
 - *La trame ne sera pas envoyée totalement et sera donc erronée (CRC faux) et rejetée par les cartes réceptrices. Elle sera ensuite réémise*

- 1. L'interconnexion des commutateurs et des concentrateurs
 - Schéma

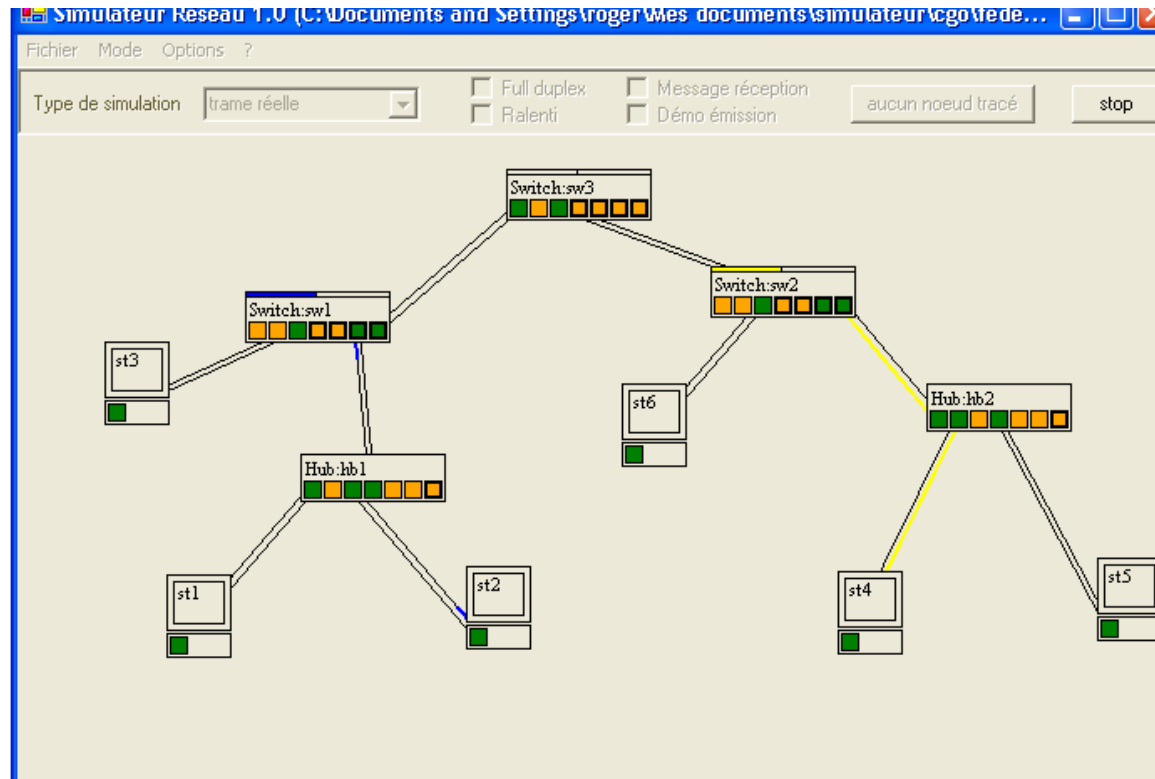


■ *Questions*

- La collision est-elle propagée ?
- Une trame en collision est-elle commutée ?
- Qu'arrive-t-il à une trame incomplètement transmise ?

■ 2. Architecture fédérée autour des commutateurs

- On utilise fédéswitch.xml
- Il s'agit de noter le résultat des actions suivantes
 - Quels sont les destinataires de la trame ?
 - Quels sont les lecteurs de la trame ?



■ 2. Architecture fédérée autour des commutateurs

– Manipulations

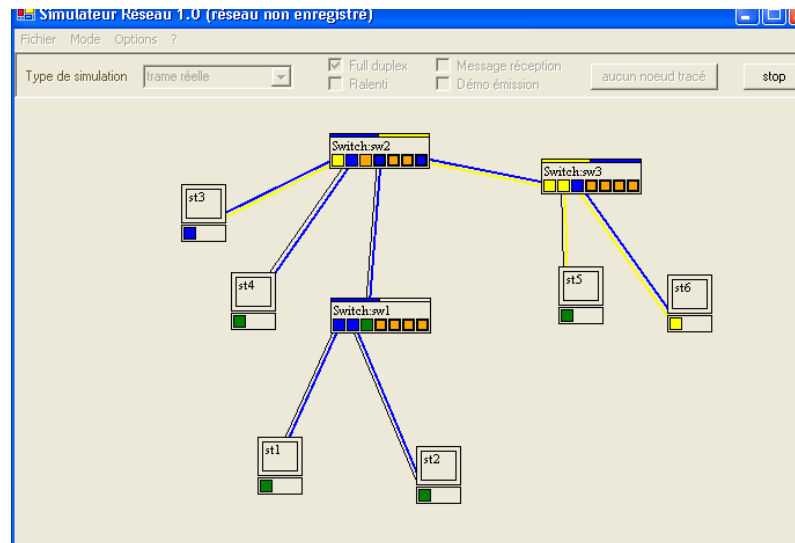
- On envoie une trame de st1 vers st2 .
- On envoie une trame de st2 vers st1.
- On sélectionne le type de simulation « trame réelle ».
- On envoie une trame de st1 vers st2 et en même temps on envoie une trame de st2 vers st1.

– Toujours en « trame réelle »

- On envoie une trame de broadcast de st1 vers st2 et en même temps une trame de broadcast de st2 vers st1
- *Malgré l'utilisation d'adresses de broadcast, la collision ne se propage pas*
- *On observe aussi que le commutateur ne stocke que les trames correctes et non les trames « en collision »*
- St3 envoie une trame à st1.
- St6 envoie une trame à st1.
- St4 envoie une trame à st1.
- St6 envoie une trame à st5.
- St3 envoie une trame à st6.
- St3 envoie une trame à st4.
- St1 et st5 envoient simultanément une trame de broadcast.

■ 3. Architecture entièrement commutée

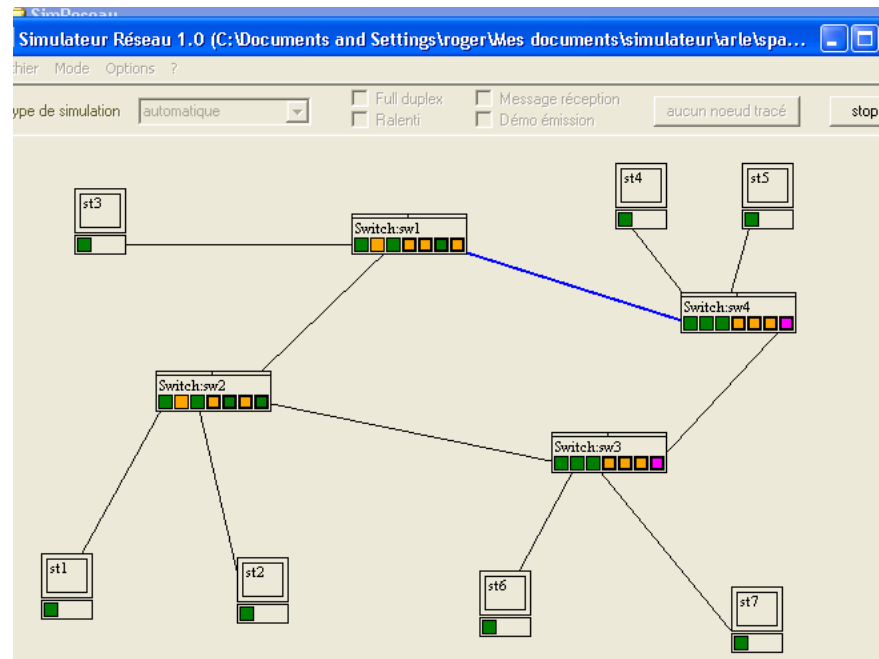
- On utilise le fichier full.xml
- Dans une architecture entièrement commutée,
 - il n'y a plus de collision on peut donc travailler en mode full-duplex car la paire de réception n'est plus monopolisée pendant l'envoi pour détecter la collision
- On peut utiliser aussi le type « on the fly »
 - Attention « on the fly » et « full duplex » sont des techniques indépendantes, on peut être à la fois en mode « store and forward » et en mode « full duplex »



■ Questions

- Quels sont les avantages et les inconvénients du type « on the fly » ?
- Quel est l'intérêt du « full duplex » ?
- Peut-on être en full-duplex sur le commutateur et pas sur la carte réseau ?

- 4. Ethernet commuté : 802.1d spanning tree (arbre de recouvrement)
 - Pour gérer la tolérance aux pannes des commutateurs, on introduit des circuits redondants qui sont inactifs en fonctionnement normal et activés en cas de panne
 - Pour cela chaque commutateur doit activer le protocole spanning tree (802.1d)
 - Les commutateurs doivent être configurés en mode conception réseau

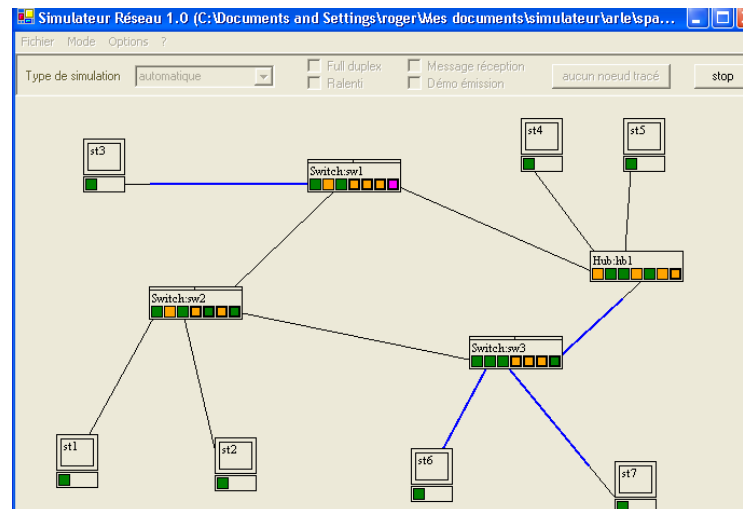


- 4. Ethernet commuté : 802.1d spanning tree (arbre de recouvrement)
 - Manipulation
 - On utilise le fichier span1.xml
 - On envoie de broadcast à partir de st1
 - Questions
 - Pourquoi un port est-il invalidé sur les switch3 et le switch4 (ces ports apparaissent par défaut en rose) ?
 - Quelle serait la conséquence d'une boucle sur la transmission de la trame de broadcast ?
 - Quelle serait la conséquence pour une trame *unicast* dont les commutateurs n'ont pas d'association pour l'adresse destinataire ?

■ Questions

- Pourquoi un port est-il invalidé sur les switch3 et le switch4 (ces ports apparaissent par défaut en rose) ?
- Quelle serait la conséquence d'une boucle sur la transmission de la trame de broadcast ?
- Quelle serait la conséquence pour une trame *unicast* dont les commutateurs n'ont pas d'association pour l'adresse destinataire ?

- 4. Ethernet commuté : 802.1d spanning tree (arbre de recouvrement)
 - Manipulation (suite)
 - On utilise maintenant le fichier span2.xml
 - Ici on a un concentrateur connecté à deux commuateurs. Bien que les concentrateurs ne gèrent pas le protocole spanning tree, ils diffusent (niveau 1 oblige) les trames 802.1d et permettent donc l'invalidation d'un circuit y compris à travers eux
 - On envoie une trame à partir de st1. *Le circuit entre sw1 et hb1 a été invalidé, il n'y a pas de boucle*



■ Questions

- Comment les deux commutateurs ont-ils pu invalider le circuit à travers le concentrateur ?

Fiche 5

■ Objectifs

- Montrer le fonctionnement des VLAN
- Montrer l'interconnexion de commutateurs gérants des VLAN de niveau 1 en utilisant des ports 802.1q

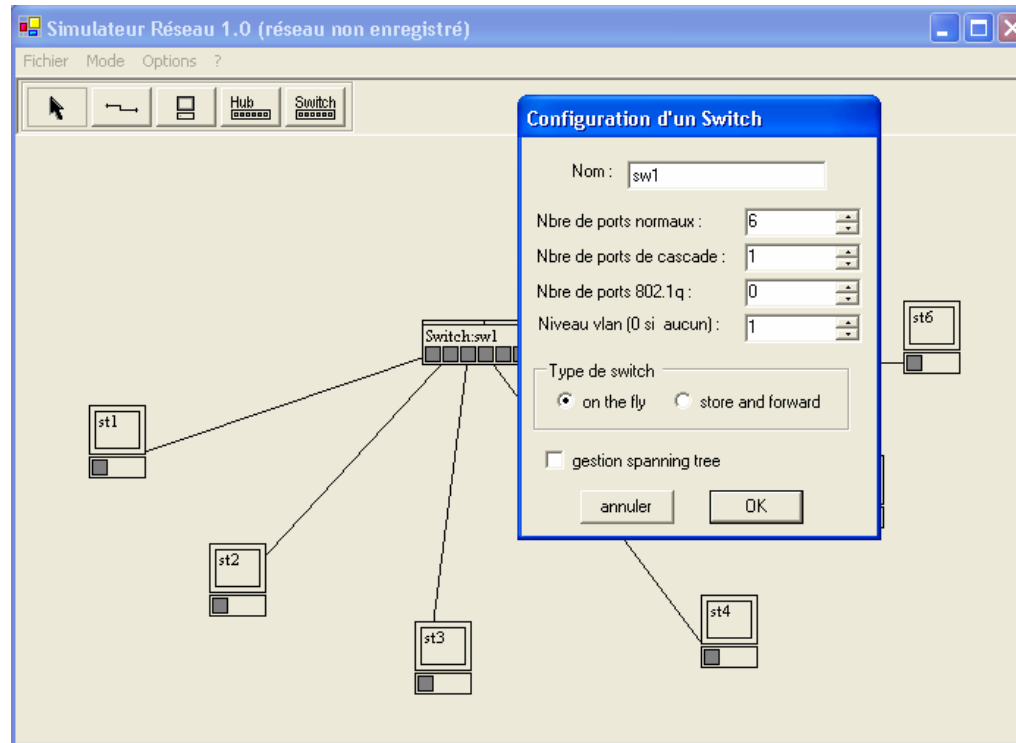
Fiche 5

■ Objectifs

- Montrer le fonctionnement des VLAN
- Montrer l'interconnexion de commutateurs gérants des VLAN de niveau 1 en utilisant des ports 802.1q

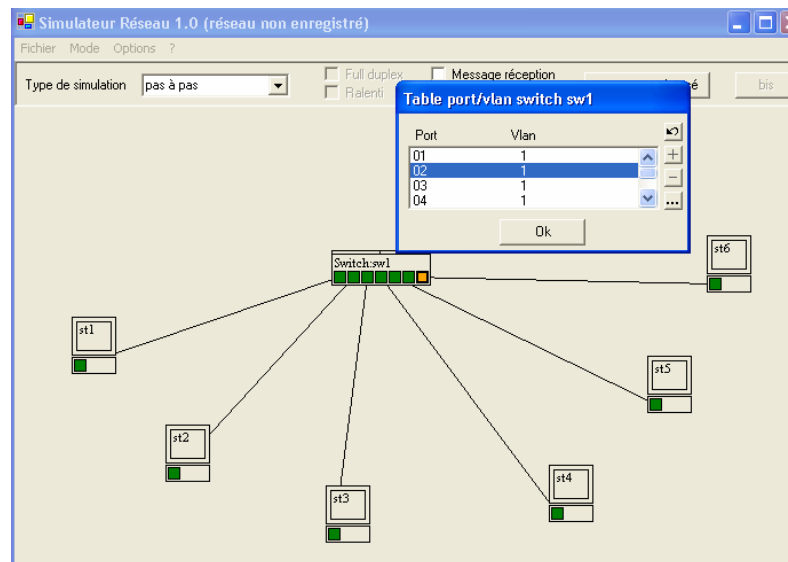
1. Comportement d'un commutateur gérant des VLAN de niveau 1

- *On utilise le fichier unswitchvlan1.xml*
- Définition du niveau de VLAN
 - En mode « conception ethernet » choisir 1 comme niveau de VLAN



– Affectation des ports au VLAN

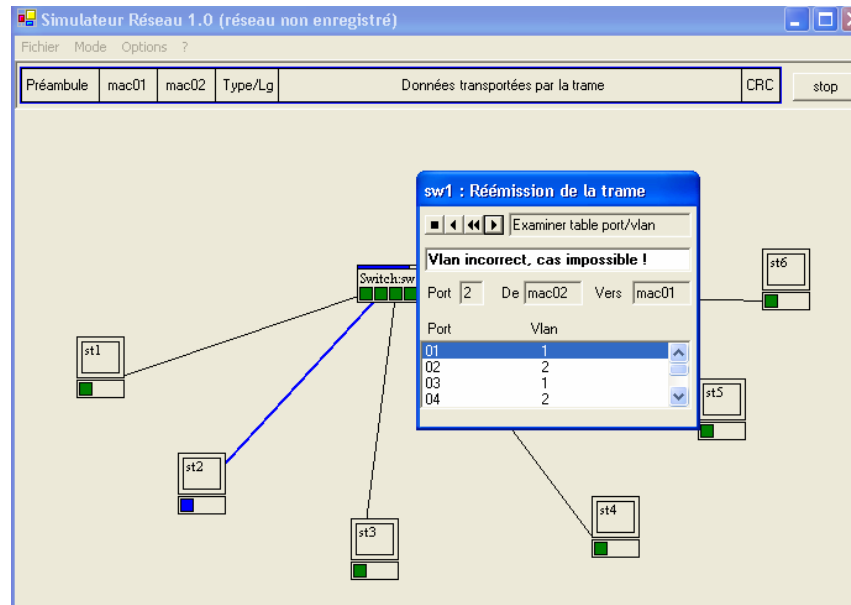
- On utilise le mode Ethernet
- Clic droit sur le commutateur, **Edition de la table port / Vlan**
 - On constate que tous les ports appartiennent au même VLAN, en l'absence de définition explicite de VLAN
- On sélectionne le port que l'on veut affecter et on clique sur le bouton avec les 3 points
- On affecte les ports 2 4 et 6 au VLAN 2; les ports 1 3 et 5 restent donc affectés au VLAN 1
- On considère que le **VLAN 1** est le **VLAN invité**
 - C'est à dire le VLAN par défaut lorsque des ports ne sont pas affectés de façon explicite à un VLAN



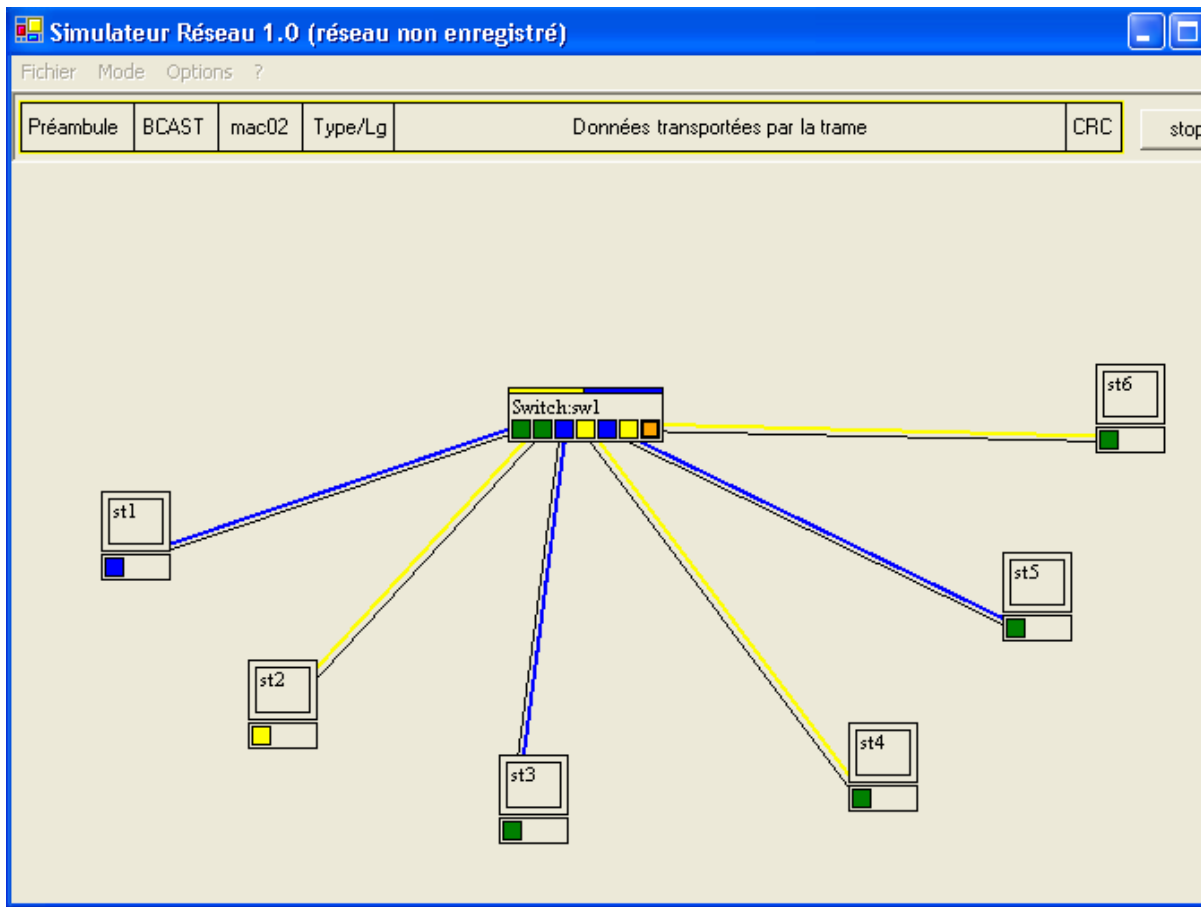
- Fonctionnement du commutateur avec le VLAN de niveau 1.
 - On envoie une trame de broadcast à partir de st1
 - *st3 et st5 reçoivent*
 - *Le commutateur a reçu une trame sur un port affecté au VLAN 1 il la transmet sur tous les ports affectés au VLAN 1*
 - On envoie une trame unicast de st5 vers st3.
 - *st3 et st1 reçoivent car le commutateur ne sait pas où se trouve st3 qui n'a pas encore émis de trame*
 - *Le commutateur a associé st1 au port numéro 1 mais il ne présuppose pas qu'il n'y a qu'un seul poste connecté au port numéro 1 (il pourrait y avoir un concentrateur ou un autre commutateur)*
 - On envoie une trame unicast de st1 vers st2
 - *St3 et st5 reçoivent la trame*
 - *St2 est sur un port affecté au VLAN 2 mais le commutateur l'ignore car st2 n'a pas encore émis de trame*
 - *Ce cas est impossible dans la réalité car un poste qui démarre enverra au minimum une trame « arp gratuit » pour tester l'unicité de son adresse IP*

– Fonctionnement (suite)

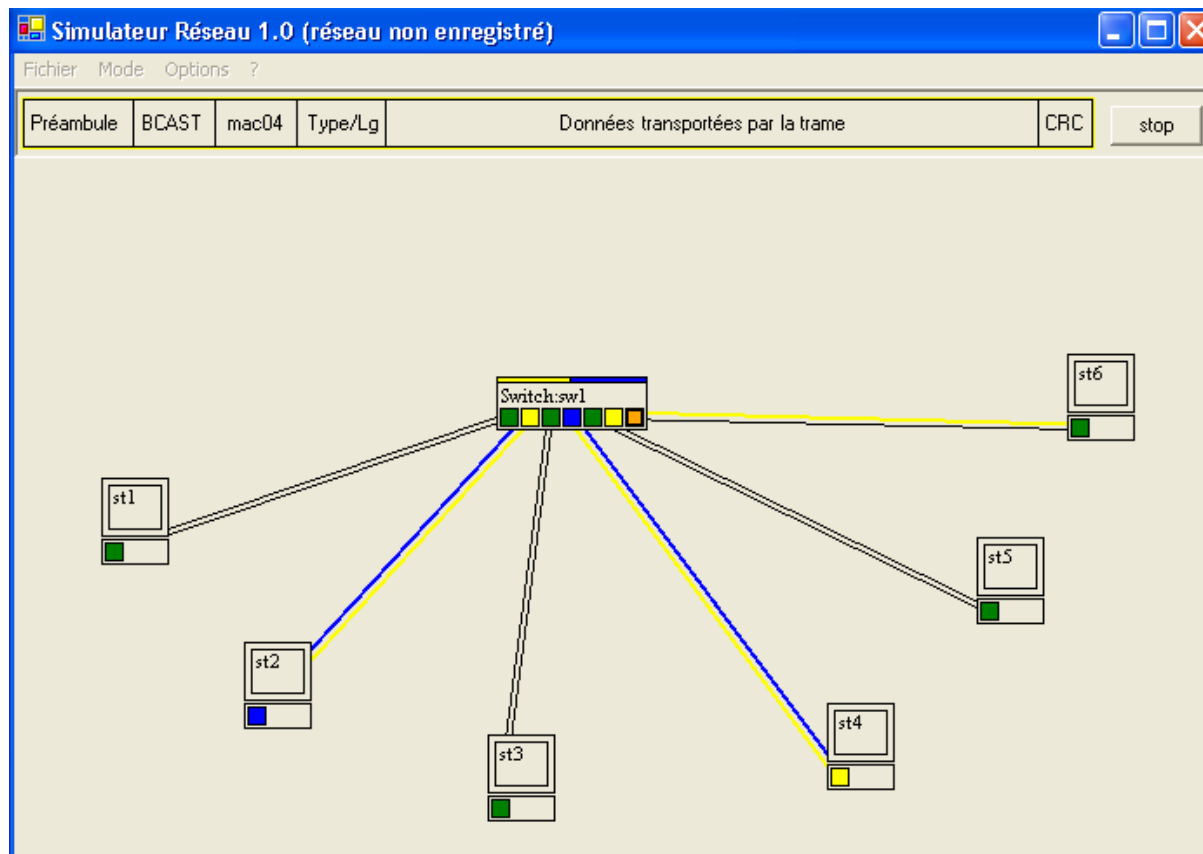
- On coche l'affichage des messages de réception
- St2 envoie une trame vers st1
 - *La trame est perdue*
 - *Elle n'est pas réémise par le commutateur car le commutateur a associé st1 au port 1 qui est affecté au VLAN 1 alors que la trame lui est parvenue par le port 2 affecté au VLAN 2*
 - *Ce cas est impossible dans la réalité car pour obtenir une adresse MAC la carte réseau a du procéder au préalable à un échange (ARP par exemple)*
- Recommencer cet envoi en demandant le « traçage » du switch. Ne pas oublier de choisir le type de simulation « pas à pas »



- Fonctionnement (suite 2)
- *Montrer les domaines de diffusion*
 - Se mettre en mode « trame réelle »
 - On envoie une trame de broadcast à partir de st1 et de St2
 - *On voit bien apparaître les deux domaines de diffusion*



- *Toujours en mode « frame réelle » , on coche la case « full duplex. »*
- *On envoie une frame de broadcast à partir de st2 et de st4*
 - *Les deux broadcast se croisent sur les paires st2 et st4 puis se succèdent sur les paires st6*
 - *St1 st3 et st5 ne sont pas concernés par l'échange*



■ *Questions*

- *Relever les destinataires de la trame*
- *Relever les lecteurs de la trame*
- *Anticiper ou Expliquer les résultats*
- *Comment le commutateur sait-il à quel VLAN est raccordée une station ?*
- *Que se passe-t-il si un port n'est pas associé à un VLAN ?*
- *Qui crée la table Vlan/ port ?*
- *Qui crée la table Mac / port ?*

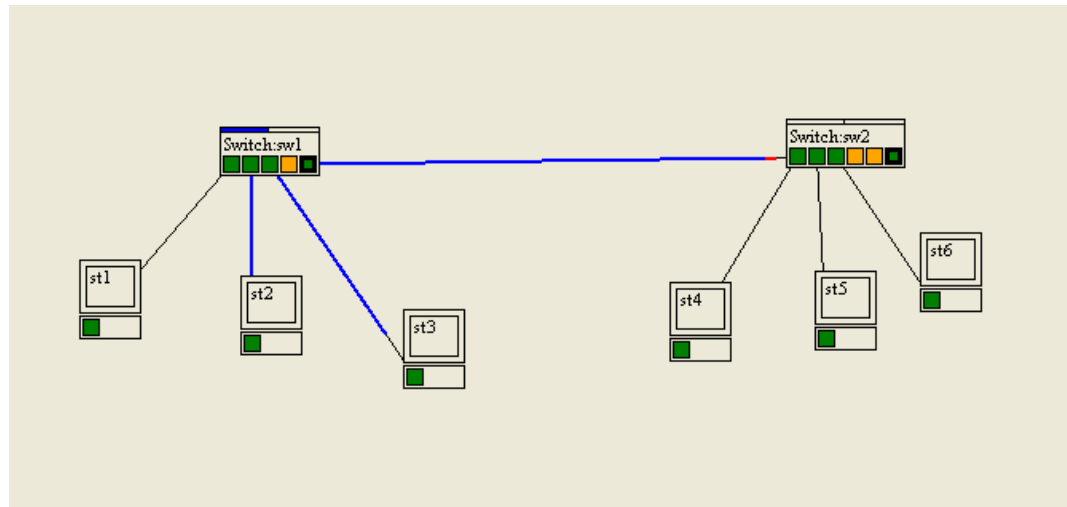
■ Questions-réponses (suite)

– *Quel rapport entretiennent les tables Mac / port et les tables Port / Vlan ?*

- *Lorsque le commutateur reçoit une trame sur un port, le commutateur parcourt la table port / vlan pour associer la trame à un VLAN puis analyse l'adresse du destinataire. Il parcourt la table MAC / port pour retrouver le port associé à cette adresse. Si le port est trouvé et qu'il appartient au VLAN de la trame on émet la trame sur ce port. Si le port est trouvé mais qu'il n'appartient pas au VLAN on détruit la trame (cas improbable dans la réalité). Si le port n'est pas trouvé on émet la trame sur tous les ports associés au VLAN de la trame.*

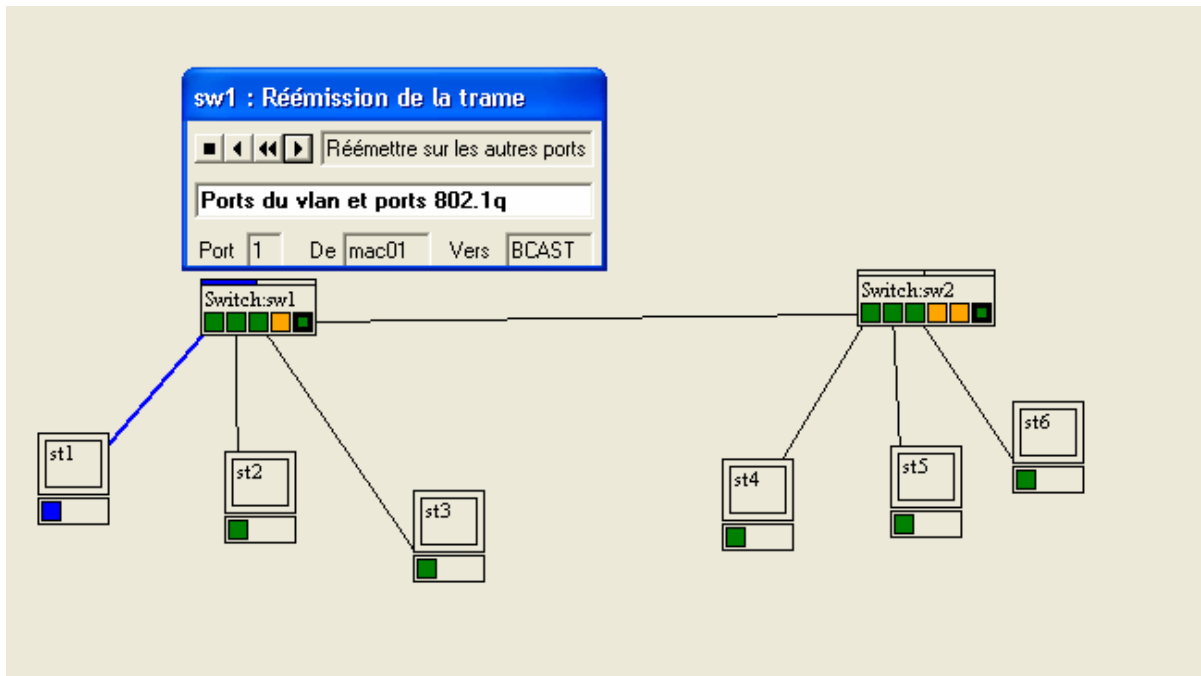
2. Gestion de deux VLAN de niveau 1 sur deux commutateurs avec 802.1q.

- On utilise le réseau deuxswitchvlan1.xml
 - Il s'agit de montrer le comportement de VLANs associés à un seul commutateur avec l'utilisation du port 802.1q
 - Le switch1 gère le VLAN1
 - Le switch2 gère le VLAN2
 - Les deux commutateurs sont interconnectés par un port 802.1q

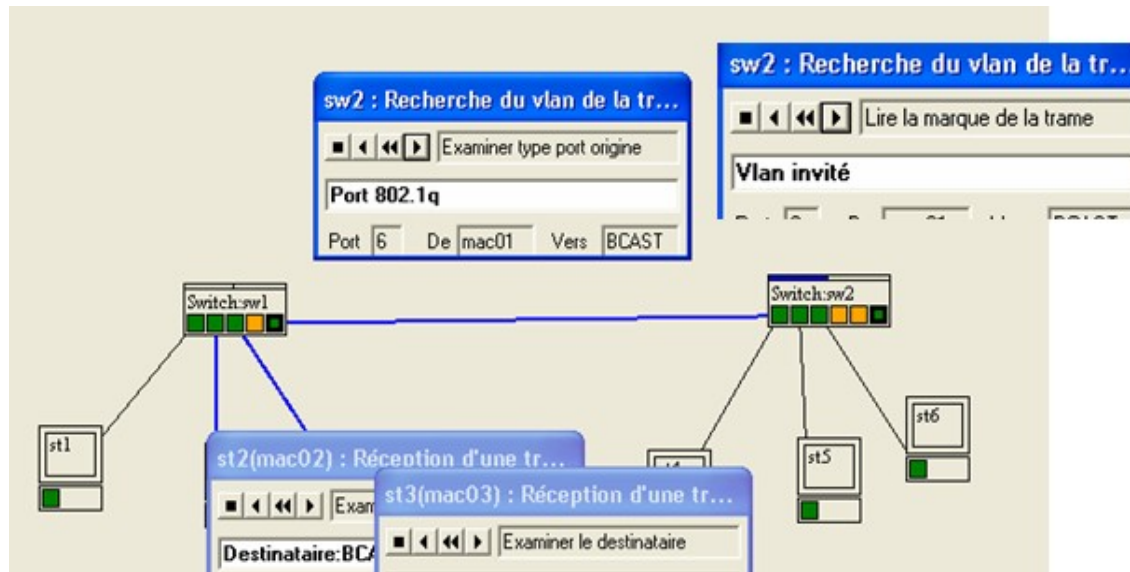


- *st1 envoie une trame de diffusion*
- *St2 et st3 reçoivent*
- *Sw2 reçoit aussi mais ne transmet pas à st4 st5 et st6*
- *Le type de simulation est automatique*
- *St2 et st3 reçoivent*
- *Sw1 transmet à sw2 par son port 802.1q en marquant la trame (petite marque rouge en début du tracé de la trame)*
- *sw2 ne sait pas par quel port la trame a été émise sur sw1 la seule façon pour lui d'associer cette trame à un VLAN est la marque (le tag) qui a été placée par le port 802.1q sur la trame*
- *En associant cette trame à un VLAN sw2 peut décider de ne pas la diffuser vers st4 st5 et st6, plus précisément il la diffuse aux ports qui sont associés au même VLAN que la marque*

- Rejouons la séquence en « traçant » les deux commutateurs. Et choisissons le type de simulation « pas à pas »

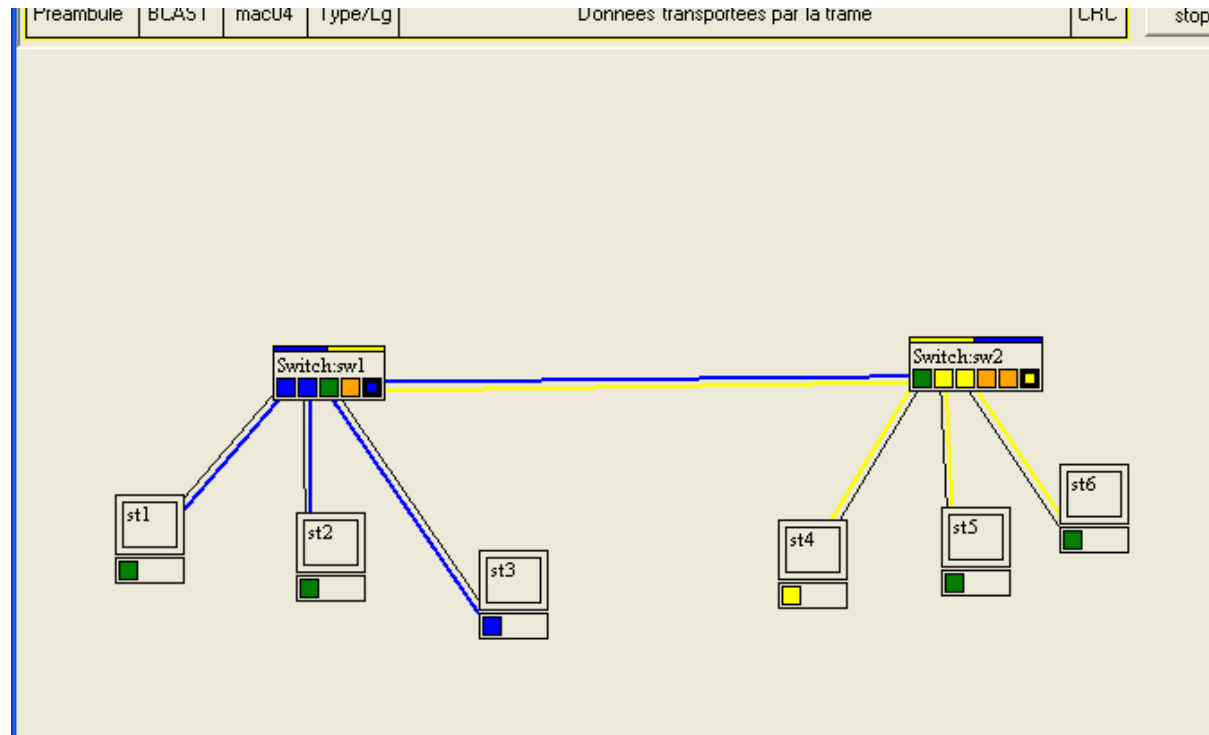


- on constate que le commutateur sw1 marque la trame et que le commutateur sw2 lit la marque de cette trame.



– Mettons nous maintenant en type de simulation « *trame réelle* ».

- st1 et st4 envoie simultanément une trame de broadcast.
Les trames sont diffusées dans chaque VLAN et sur les ports 802.1q après avoir été marquées
- *Chaque commutateur associe ainsi la trame à un VLAN et peut prendre la bonne décision concernant son acheminement*

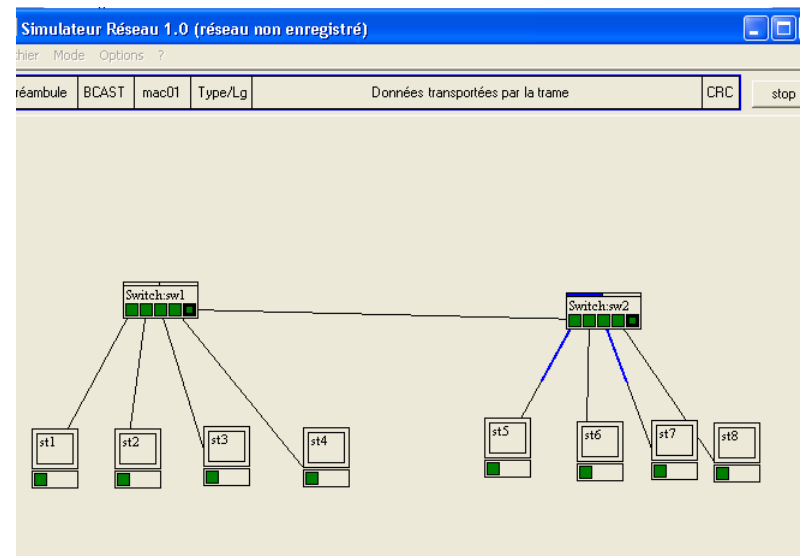


■ Questions

- Relever les destinataires de la trame
- Relever les lecteurs de la trame
- Anticiper ou Expliquer les résultats
- Un port 802.1q est-il affecté à un VLAN ?
- Qu'est-ce qu'un lien « trunk » (trunk link) ?
- Pourquoi le marquage de la trame est-il indispensable ?
- Les cartes réseaux des postes lisent-elles la « marque » 802.1q ?
- Une carte réseau peut-elle se trouver sur un lien « trunk link » ?

■ Répartition d'un VLAN de niveau 1 sur plusieurs commutateurs

- On utilise le fichier deuxswitchvlan1réparti.xml
- Il s'agit de montrer le comportement de VLAN répartis sur plusieurs commutateurs.
 - Ici les VLAN 1 et 2 sont répartis sur sw1 et sw2
 - Sur sw1 les stations 1 et 3 sont sur le VLAN 1 et les stations 2 et 4 sur le VLAN 2
 - Sur sw2 les stations 5 et 7 sont sur le VLAN 1 et les stations 6 et 8 sur le VLAN 2
 - Cela donne respectivement st1 st3 st5 et st7 sur le VLAN 1.
 - St2 st4 st6 et st8 sont donc sur le VLAN 2



– Manipulation

- On se met tout d'abord en « type de simulation automatique »
- st1 envoie une trame de broadcast
 - *St3 st5 et st7 reçoivent la trame*
 - *Sw1 a transmis une trame marquée à sw2. sw2 a lu la marque et a commuté la trame vers les bons ports*
- St5 envoie une trame unicast vers st1
 - *st1 reçoit la trame*
 - *Sw2 a transmis uniquement sur le port 802.1q auquel il a associé st1 (on peut éditer la table mac/port) Il marque la trame avec le VLAN du port émetteur soit le VLAN 1*
- *sw1 récupère la trame, lit la marque, lit l'adresse du destinataire et commute sur le bon port à l'aide de sa table mac/port*
- St6 envoie une trame unicast vers st1
 - *Sw2 ne connaît pas le VLAN du port sur lequel est connecté st1, il transmet donc à sw1 après avoir marqué la trame avec le VLAN de l'émetteur soit le VLAN 2*

■ Questions

- Comment le deuxième commutateur associe-t-il la trame à un VLAN ?
- Les deux commutateurs s'échangent-ils les tables port / vlan ?

■ Manipulations sur vos ordis

- On est passé à Ipv6
- Ouvrir une fenêtre Cmd
- Ipconfig
 - Mon adresse IP : 192.168.1.88
 - Le masque est 255.255.255.0
 - 88 est l'adresse de l'ordi au moment de la connexion
 - Passerelle par défaut : c'est mon premier routeur, par ex. votre box. Généralement, on utilise le .1
- tracert (eq. Traceroute sur linux) www.google.fr
 - Donne tous les routeurs intermédiaires avec le temps pour accéder aux adresses IP, sauf s'ils sont protégés (affichage de *)